Technical Report

Deployment and Best Practices Guide for Clustered Data ONTAP 8.1 Windows File Services

Bingxue Cai, NetApp

September 2011 | TR-3967

Executive Summary

File services are an essential part of every customer’s storage environment. NetApp® storage systems deliver highly reliable file services that use the Common Internet File System (CIFS) protocol. This document describes the CIFS infrastructure in clustered Data ONTAP® and introduces best practices and diagnostic processes during feature setup.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

NetApp clustered Data ONTAP 8.1 storage systems deliver highly reliable and scalable file services through the Server Message Block (CIFS/SMB) protocol. This document presents an overview of Server Message Block (SMB) functional modules, describes each SMB feature and configuration with best practices, and offers a selection of error diagnoses and solutions. It also contains information related to feature deployment and SMB statistics.

1.1 Audience

This document is for IT administrators, solution architects, technical architects, professional service engineers, and system engineers.

1.2 Purpose and Scope

This document describes the best practices for Windows® File Services in clustered Data ONTAP and provides a high-level view of how CIFS functionality is implemented in a clustered storage system. Specifically, this document discusses:

- CIFS architecture in clustered Data ONTAP
- CIFS features and best practices
- Diagnosis and tracing
- SMB statistics

For best practices relating to the general configuration of a cluster, refer to Data ONTAP 8.0 Cluster-Mode Implementation Guide.

For information about the Microsoft® services and features discussed in this paper, go to www.microsoft.com.

For information about feature details and services procedures on NetApp storage systems, see the "Data ONTAP File Access and Protocol Management Guide," available from the NetApp Support site (formerly NOW®).

2 Overview of Windows File Services in Clustered Data ONTAP

A cluster is a scalable system that allows you to group pairs of controllers, share physical resources, and distribute workloads across the systems while consolidating management for administrators into one unified interface. The clustered Data ONTAP 8.1 storage system has the capability to join up to 24 individual nodes or 12 pairs of nodes.

When CIFS is enabled within a cluster, each CIFS server runs on top of a virtual server (Vserver), which is a fully virtualized storage appliance. Although only one license is required for the entire cluster, CIFS must be enabled for each Vserver where CIFS is desired.

A Vserver consists of volumes that are associated within a Vserver's namespace and can be presented as a single CIFS share to the clients. The namespace presented by a Vserver can change by manipulating the volumes through their junctions or mounting relationships.

Additionally, Vservers allow a single authentication domain for each protocol. This mechanism makes sure that the data that is visible on a given Vserver cannot be accessed from any other Vserver or CIFS server.

Vservers also need virtual interfaces (VIFs) to send and receive requests from the network. VIFs are a set of IP addresses assigned to network ports associated with a Vserver along with VIF failover rules. The failover rules allow the cluster to reroute network traffic to other network ports when the original VIF has incidents of port or port aggregate failure.

CIFS server setup and configuration can be accessed from any node in a cluster. Management of a given CIFS server can be delegated to an administrator who does not have detailed knowledge of the cluster configuration or the exact location of the Vserver. Combined with secure separation of data access, this management model allows the CIFS server to serve customers by using a storage service provider (SSP) model, where each “customer” of the SSP manages his or her data separately and securely from the others.

2.1 Windows File Services Architecture

CIFS architecture in clustered Data ONTAP has been drastically changed from the traditional architecture used in Data ONTAP 7 and in Data ONTAP 8.1 operating in 7-Mode. The entire CIFS infrastructure has been reconstructed for scalability and robustness in the cluster organization.

A CIFS server consists of four functional modules that interact through remote procedure calls (RPCs):

- **CIFS configuration.** All CIFS configuration is done through a command line interface or through System Manager. When the administrator creates or changes the CIFS configuration, the configuration data is populated to the whole cluster.
- **Networking.** Networking blade (NBlade) funnels all client-facing and cluster traffic. It is responsible for processing CIFS/SMB requests, communicating with other CIFS functional modules for further processing, and generating the CIFS/SMB responses.
- **Storage.** Disk blade (DBlade) is responsible for processing file operation requests sent by NBlade. DBlade receives configuration information related to volumes, aggregates, Snapshot™ copies, qtrees, logical mirrors, and so on from other modules.
- **Security service.** The security service module provides multifunctional, highly scalable, and reliable security services to the whole cluster system. It also has its own log (`/mroot/etc/mlog/secd.log`) for diagnosis and security tracing. Performance and functionality of the security service module depend heavily on external servers, including Windows domain controllers, DNS servers, NIS servers, and LDAP servers. This module offers the following features for the whole cluster system:
  - Authentication

    NTLMv1, NTLMv2

    Kerberos

    Microsoft HiSec security
  - Name mapping
  - Automatic discovery of Microsoft Active Directory® services
  - Caching mechanisms

2.2 CIFS Features in Clustered Data ONTAP 8.1

This section contains a comprehensive feature list and an overview of the supported CIFS features in clustered Data ONTAP 8.1.

- CIFS/SMB and SMB 2.1
- Active Directory site awareness and optimization
- Automatic discovery of the Active Directory services with optimization
- NTLMv1, NTLMv2, and Kerberos authentication
- SMB client-side signing for Signed SMB domain controller connection
- Encrypted SChannel MSRPC security for Netlogon

- Microsoft HiSec security support
- DFS referral through widelink and NFS symbolic link (symlink) resolution
- Home directories
- Name mapping using RDB and LDAP
- CIFS shares management
- Credential cache
- Extensive caching of connections to external servers
- Extensive caching of all data from external servers
- Group Policy Object support
- Quotas

3 CIFS Server Creation

The CIFS servers must be created on the cluster system before the cluster system can begin to host the CIFS requests. You can create up to 256 CIFS servers per cluster in clustered Data ONTAP 8.1. This section details the CIFS server creation and feature configuration, along with best practices. Other features and their best practices are discussed in the sections that follow.

3.1 Prerequisites

This section describes prerequisites that are necessary before creating a CIFS server.

1. A valid CIFS license.
2. A valid cluster.

   The cluster can be created through a setup wizard by using one of the following commands:

```
cluster::> cluster setup
```

   Or

```
cluster:: cluster create -license <license> -clustername <cluster_name> -ipaddr1 <IP Address> -
ipaddr2 <IP Address> -netmask <IP Address> -mgmt-ip <IP Address> -mgmt-netmask <IP Address>  -
mgmt-gateway <IP Address> -mgmt-port <netport>
```

3. A valid Vserver with a data volume.

   The Vserver can be created through the setup wizard by using one of the following commands:

```
cluster::> vserver setup
```

   Or

```
cluster::> vserver create -vserver <vserver> -rootvolume <volume> -aggregate <aggr>
-ns-switch file,nis,ldap -language en_US -rootvolume-security-style mixed
```

4. Network interface with routing information.
   a. Make sure that the network interface for the management port and the corresponding route are properly set. To create a network interface for the management port and the corresponding route, use the following commands:

```
cluster::> network interface create -server <vserver> -lif <logical interface> -role node-mgmt -
home-node <nodename> -home-port <admin port> -address <IP Address>  -netmask <IP Address>
cluster::> network routing-group route create -server  <vserver>
 -routing-group <Routing Group> -gateway <gateway IP>
```

   b. Make sure that the network interface for the data port and the corresponding route are properly set. To create a network interface for the data port and the corresponding route, use the following commands:

```
cluster::> network interface create -server <vserver> -lif <logical interface> -role data -home-
node <nodename> -home-port <data port> -address <IP Address> -netmask <IP Address>
cluster::> network routing-groups route create -server <vserver> -routing-group <Routing Group> -
gateway <gateway IP>
```

5. Enable and properly configure DNS to provide the domain name services (DNS). The following command creates a DNS entry for a specific domain with a list of DNS server IPs:

```
cluster::> vserver services dns create -vserver <vserver> -domains <FQDN_Of_Domain>
-name-servers <List_of_IP_Of_DNS_Servers>
```

6. Name the service switch and setup.
   – If the name service is set to local file only, make sure that the local UNIX® users and/or groups are created by using the following command:

```
cluster::> vserver show -vserver <vserver> -fields ns-switch
vserver   ns-switch
-------   ---------
<vserver> file

cluster::> vserver services unix-user create -vserver <vserver> -user <username> -id <user id> -
primary-gid <group id>

cluster:: vserver services unix-group create -vserver <vserver> -name <groupname> -id <group id>
```

   – If the name service is set to NIS, configure the NIS service accordingly by using the following command:

```
cluster::> vserver show -vserver <vserver> -fields ns-switch
vserver     ns-switch
-------     -------------
<vserver>   nis

cluster::> vserver services nis-domain create –vserver <vserver>  -domain <NISdomain>
-servers <IP_of_NIS_server>
```

   – If the name service is set to LDAP, configure the LDAP service accordingly by using the following command:

```
cluster::> vserver show -vserver <vserver> -fields ns-switch
vserver     ns-switch
-------     -------------
<vserver>   ldap
```

To configure the LDAP service:

a. Copy the predefined read-only LDAP schema to your own modifiable schema. Data ONTAP provides two default schemas:

```
cluster::> vserver services ldap client schema show
Schema Template Comment
--------------- ---------------------------------------------------------------
AD-SFU          Schema based on Active Directory Services for UNIX (read-only)
RFC-2307        Schema based on RFC 2307 (read-only)
2 entries were displayed.
```

You can adjust your schema to match the one installed on the targeting LDAP server. If the LDAP server is Active Directory, copy the AD-SFU schema; if the LDAP server is an OpenLdap/RFC2307-based server, copy the RFC-2307 schema.

```
cluster::> vserver services ldap client schema copy -schema AD-SFU MyLdapSchema
```

b. Create the LDAP client that binds to the schema that you created in the previous step.

If the targeting LDAP server is Active Directory:

```
cluster::> vserver services ldap client create -client-config <Client Config Name>
-ad-domain <Domain>-schema MyLdapSchema
```

If the targeting LDAP server is an OpenLdap/RFC2307-based server and the LDAP server does not accept anonymous login, you must specify bind-dn and bind-passwd for authentication purposes. You must also specify base-dn and base-scope to define and/or narrow down the search within the base and the scope.

```
cluster::> vserver services ldap client create -client-config <Client Config Name>
-servers <IP Address,>-schema MyLdapSchema
```

c. Create an LDAP client for the target Vserver. You can adjust the LDAP client schema according to the one installed on the connecting LDAP server.

```
cluster::> vserver services ldap create -vserver <Vserver> -client-config
<Client Configuration Name> -client-enabled {true|false}
```

7. Name mapping. Name mapping is a key feature that governs the multiprotocol file access syntax. Name mapping allows flexibility in mapping the user between CIFS and NFS, but it must be used carefully because it can create confusing scenarios.

```
cluster::> vserver show -vserver <vserver> -fields nm-switch
vserver        nm-switch
---------      ---------
<vserver>      file,ldap
```

– Local name mapping. By default, a UNIX user is mapped to a Windows account with the same user name from the accessing domain, and vice versa.

```
cluster::> vserver name-mapping create -vserver <vserver> -direction win-unix position <position>
-pattern domain\\NTUser -replacement UnixUser
cluster::> vserver name-mapping create -vserver <vserver> -direction unix-win -position
<position> -pattern UnixUser -replacement domain\\NTUser
```

– LDAP name mapping. If LDAP name mapping is chosen, the LDAP name mapping schema must be adjusted according to the schema installed on the targeting LADP server.

8. Export policy. Create an export policy for server access and make sure that CIFS is included in the access protocol. The default export policy allows CIFS access.

9. Time synchronization. For Kerberos authentication to work, the cluster and the Kerberos server (Active Directory by default in a CIFS environment) clocks need to be within 5 minutes of each other. You can either manually adjust the date and time of the installer node or you can set the NTP server for time synchronization.

```
cluster::> system services ntp server create -node <node> -server <IP Address>
cluster::> system services ntp config modify -enabled true
```

10. Obtain the Fully Qualified Domain Name of the installing domain.

11. Obtain the name and password for a domain user who has administrator rights.

3.2 Preferred Domain Controllers (Optional)

The storage server automatically discovers Active Directory services during server setup and afterward based on various criteria such as the setup on the storage server, the Active Directory site setup, the distance from the storage server to the targeting services, and so on.

The storage server administrator can explicitly set up a list of Windows DCs to connect to as its top preference for various CIFS operations. to set up the preferred DCs, use this command:

```
cluster::> vserver cifs domain preferred-dc create -vserver <vserver> -domain
<Fully Qualified Domain Name> -preferred-dc <InetAddress>, ...
```

3.3 CIFS Setup

CIFS setup in clustered Data ONTAP is done by using this command:

```
cluster::> vserver cifs create –vserver <vserver> -cifs-server <NetBIOS Name> -domain
<Fully Qualified Domain Name>
```

Example:

```
cluster::> vserver cifs create -vserver vSrv1 -cifs-server cifsSrv1 -domain cifsdom.netapp.com
```

After a successful setup, the following entry appears in the CIFS table:

```
cluster::> vserver cifs show
           Server          Domain/Workgroup      Authentication
Vserver    Name            Name                  Style
--------   --------------  --------------------  ----------------------------
vSrv1      CIFSSRV1        CIFSDOM               domain
```

3.4 CIFS Server Validation

After a successful CIFS setup, you can run a few more commands to simulate the user login and user mapping processes to validate the proper CIFS server setup.

To access those commands, you need to be in diagnostic mode:

```
cluster::> set diag
Warning: These diagnostic commands are for use by NetApp personnel only.
Do you want to continue? {y|n}: y
```

1. Simulate user login:

```
cluster::*> diag secd authentication login-cifs -node <node_name> -vserver <vserver> -user
<domain>\<username>
```

2. Get the CIFS user's credential, including its mapping UNIX identity:

```
cluster::*> diag secd authentication show-creds -node <nodename> -vserver <vserver>
-win-name <domain>\<username>
```

3.5 Working with Pre-Windows 2000 Compatible and Noncompatible Domains

Starting with Windows Server® 2003, the Windows Authorization Access (WAA) group was introduced to simplify the process of granting read access on the token groups global and universal (TGGAU) attribute. Members of the WAA group have access to the computed TGGAU attribute on user objects.

When the domain is in pre-Windows 2000 compatibility access mode, the Everyone group has read access to the TGGAU attribute on user account objects and on computer account objects. In this mode, clustered Data ONTAP CIFS servers have access to TGGAU.

When the domain is not in pre-Windows 2000 compatibility access mode (for example, if you have newly installed Windows 2003 or Windows 2008 domains that are not in pre-Windows 2000 compatibility access mode), you must manually add clustered Data ONTAP CIFS servers to the WAA group to access TGGAU.

3.6 Server Management with Microsoft Management Console

Domain administrators can connect Microsoft Management Console (MMC) or Computer Management to clustered Data ONTAP 8.1 to monitor and view SMB shares, active sessions, and open files. Domain administrators cannot modify, delete, or create the SMB share definition, terminate SMB sessions, or close open files through MMC with this release.

3.7 Common Failure and Solutions

Because of the complexity of Active Directory and Windows file server environments, it is not uncommon for the CIFS setup to fail. This section describes the most common mistakes and errors you might encounter during the CIFS setup and their corresponding diagnoses and solutions.

Kerberos Time Skew

When the storage server's clock is not in sync within 5 minutes of the domain controllers, the authentication service through Kerberos stops working. The security service logs the time skew error messages. To check the event log, see section 10.1, "Event Logging", in Appendix B.

Adjust the cluster timer by using the following command:

```
cluster::> date [[[[[cc]yy]mm]dd]hhmm[.ss]]
```

Or you can set up the Network Time Protocol (NTP) server to synchronize the cluster clock with the target server.

Best Practice

Kerberos has strict time requirements, which means that the clocks of the involved hosts must be synchronized within configured limits. The cluster timer must be in sync with the Windows domain controller to avoid time-related problems. To set up the NTP server, use this command:

```
cluster::> system services ntp server
```

DNS

- **Configuration on the cluster.** If DNS is not configured, not enabled, or configured incorrectly, the security service reports a configuration-related error during CIFS server creation.
- **Configuration in the Windows domain.** If DNS contains no records for DCs in the CIFS domain, the security service finds no DCs to use during CIFS server creation.

Unable to Create a CIFS Server Due to Authentication Issue

If a non-administrator user account was given for use during CIFS server creation, the server creation process gets an Access Denied error from the domain controller.

Best Practice

Be sure to create a CIFS server with the domain administrator login.

Unable to Create a CIFS Server Due to Networking Ossue

Check the network setup and make sure that:

- You have the correct data-LIF (Logical Interface) setup for the Vserver
- You have the corresponding routing groups for the data-LIF created
- The routing groups contain the correct routes to the Windows domain controllers

Unable to Create a CIFS Server Due to Export Policy Restriction

Best Practice

Before a CIFS server is created, make sure that the export policy rule includes the CIFS protocol.

To Create a CIFS Server Due to Vserver Restriction

Best Practice

During Vserver creation (especially if you use the wizard to create the Vserver), make sure that the CIFS protocol is included in the protocol list.

Slow Performance Due to Incorrect Windows Site Configuration

When Windows sites are not correctly configured, the security service may attempt to use a remote DC for authentication and authorization; this could slow the overall CIFS performance.

Authentication Errors

Account Cannot be Authenticated into the CIFS Server

Table 1) Account authentication error.

Error	Cause and Resolution
The authentication request failed with DC error 0xc000006a.	Log-in failure due to misspelled or wrong password.
The authentication request failed with DC error 0xc0000072.	The user login account has been disabled by the administrator.
The authentication request failed with DC error 0xc0000064.	The user does not exist in the Windows domain.
The authentication request succeeded with no DC error; however, the client still gets an authentication failure.	Name mapping is not defined, or it is not defined appropriately for the login user. The security service is unable to locate the mapping between the Windows user and the UNIX user, or the security service is unable to identify the mapped UNIX user.

NTLM Authentication Issue

Table 2) NTLM authentication error.

Error	Cause and Resolution
The Windows host is unable to authenticate with the CIFS server using the server IP address.	The connection between the CIFS server and the DC might be down; NTLM-style authentication cannot be performed

Kerberos Authentication Issues

Table 3) Kerberos authentication error.

Error	Cause and Resolution
Out-of-sync timer.	If the cluster timer is not synchronous with the domain controllers, try to adjust the cluster clock; or try to set up the NTP service according to item 9 in section 3.1.
Machine account password is out sync between the DC and the CIFS server.	Reset the password by using the following command: `cluster::> vserver cifs password-reset –vserver <vserver>`

Authorization Errors — Problems with File Access

Table 4) Authorization errors.

Symptom	Cause and Resolution
Missing NIS account for nobody or root can cause various access errors; check the event log.	Add UNIX accounts called nobody and root.
The account may be mapped to a different identity that does not have access permission.	Adjust the name mapping entries according to section 5.
The account does not have proper share-level access.	Give proper share-level permission to the account according to section 4.3.

The account does not have proper access permission to the files.	Assign the account with proper file permission for access.
The account is unable to access the files because the CIFS Server is unable to read the global and universal group information.	The CIFS server does not have proper permission to access the token group global and universal attribute. To resolve this, put the CIFS server in the Windows Authorization Account Group.

Best Practice

If you are working with Windows 2003 domain or later, always add the CIFS server to the Windows Authorization Account Group.

Cannot Map a CIFS Share

Table 5) CIFS share mapping error.

Error	Cause and Resolution
The account does not have proper share-level access.	Give proper share-level permission to the account according to section 4.3.
The account does not have proper share-level access on the target share.	Give proper share-level permission to the account according to section 4.3.
The account does not have proper access permission to share the directory.	Assign the account the proper file permission for access.

EMS Message Spamming on the Console

Table 6) Configuration errors.

Error	Cause and Resolution
Configuration errors occur frequently.	Check the event log and other logs to determine what configuration is causing the error and fix it accordingly.

4 Share Management

4.1 Share Creation

After a CIFS server is created, CIFS shares need to be created to allow file access. By default, a share is created with oplock and change notify capabilities, and the share is browsable.

To create CIFS shares, use the following command:

```
cluster::> vserver cifs share create –vserver <vserver> -share-name <share> -path <directory path
to be exported through CIFS share>
```

- To create a share that can show the Snapshot directories, during share creation, specify:

```
share-properties oplocks,browsable,changenotify,showsnapshot
```

- To create a share that acts as a home directory, during share creation, specify:

```
share-properties oplocks,browsable,changenotify, homedirectory
```

- To create a share that caches the file attributes in the NBlade for SMB access, specify.

```
share-properties oplocks,browsable,changenotify, attributecache
```

To control the lifetime of cache entries for that share when caching is enabled, specify:

```
attribute-cache-ttl value during share creation, by default the attribute cache ttl is set to
10sec
```

Best Practice

For metadata-intense CIFS traffic, this caching mechanism increases the cluster CIFS performance by reducing the traffic between NBlade and DBlade in a large cluster environment.

- To create a share that can resolve NFS symlinks, specify:

```
symlink-properties enable
```

- Specify a file or directory mode creation mask for a share through the following two properties:
  - `-file-umask`
  - `-dir-umask`

Here is an example that shows a share with properties through the `show` command:

```
cluster::> vserver cifs share show -instance

                     Virtual Server: vs1
                              Share: root
           CIFS Server NetBIOS Name: CIFSSRV
                               Path: /
                   Share Properties: oplocks
                                     browsable
                                     changenotify
                                     attributecache
                 Symlink Properties: -
            File Mode Creation Mask: -
       Directory Mode Creation Mask: -
                      Share Comment: -
                          Share ACL: Everyone / Full Control
      File Attribute Cache Lifetime: 10s

1 entry was displayed.
```

4.2 Share Validation

To check whether the newly created share can be connected and mapped, from a Windows client, issue the following commands at the command prompt:

1. The `net view` command displays a list of resources being shared on a storage server:

```
C:\> net view \\cifs_server
C:\> net view \\cifs_server_ip_address
```

2. The `net use` command connects a computer to a shared resource or disconnects a computer from a shared resource:

```
C:\> net use * \\cifs_server_name\share_name  /user:<domain_name>\<user_name> <password>”
```

4.3 Share-level Access Control Lists

By default, a share is created with `Everyone Full Control` access, which gives all users in the domain and trusted domains full access to the share. You can modify the share-level access control list (ACL) to restrict permissions for the share.

```
cluster::> vserver cifs share access-control modify -vserver <vserver>  -share <Share>
-user-or-group <User/Group Name> -permission <The access rights listed in an Access Control
Entry>
```

Example: To modify a share-level ACL to set the `Everyone Read only` control:

```
cluster::> vserver cifs share access-control create -vserver <vserver> -share testshare
-user-or-group everyone -permission Read
```

To validate the access controlled by this share-level ACL:

1. On a Windows client, map a share by issuing the `net use` command, or run `Map Network Drive` from Windows Explorer.
2. The user should only have read access on all files and directories under the mapped share. The user should not be able to create a file or directory, delete a file or directory, modify a file, and so on.

4.4 Home Directories

Home directories have been deployed most extensively in CIFS environments. A CIFS home directory is a dynamic share in memory that maps to a specific path on a virtual server that serves as the home directory for a specific user. Data ONTAP finds this path automatically by using the patterned share name and search paths that you provide.

Home directory is a special type of share that differs from other static shares in following ways:

- You cannot change the share-level ACL and the comment for a home directory.
- The CIFS `shares` command does not display the home directories.
- A home directory share name can be a patterned name.
- The share path for home directory is a relative path, while the share path for a regular static share is an absolute path

The home directory feature is not supported by Microsoft, and Microsoft can make changes to Windows that can break or damage CIFS home directories. To avoid this situation, three steps are required for a properly configured home directory:

Three steps are required for a properly configured home directory:

1. Add a valid home directory search path. A search path is the full path on the virtual server in which the storage server searches for home directories. For example, if the search path is*`/users`*, the storage server searches for the home directories only within that path. You can specify multiple

search paths for the storage server to search. Those search paths are ordered; the position value indicates the path search order for the home directory.

Example:

```
cluster::> vserver cifs home-directory search-path add -vserver vs0 -path
/home
```

```
cluster::> vserver cifs home-directory search-path add -vserver vs0 -path
/home1
```

```
cluster::> vserver cifs home-directory search-path show
```

```
Virtual
Server      Position   Path
-----------   -----------   -----------------
vs0           1              /home
vs0           2              /home1
2 entries were displayed.
```

2. Create a Home Directory share. A Home Directory share is created by using the `vserver cifs share create` command by providing a valid home directory syntax, which consists of two values:
   a. **A patterned share name.** A *share name* is the name of the share on the virtual server to which the user is trying to connect. For example, if user1 is trying to connect to \\vserver\~user1, the share name is ~user1. You can use a patterned name to specify the home directory share name. For example, if NTUser1 is trying to connect to \\vserver\~NTUser1, the patterned share name is ~%w.
   b. **A patterned share path.** A *share path* is a directory path that a CIFS share is mapped to. The path for home directory share is a relative path that is attached to the home-directory search paths (created in the previous step) to form complete share paths.

Note: Static share creation accepts only an absolute path, whereas home-directory share accepts only a relative path.

The following share patterns are supported:

– %d. User's domain

– %w. User's Windows login name

– %%. Represents a literal "%"

The following table gives examples of creating a Home Directory share, with the assumption that the search path is set to `/home`, the user's login domain is `NTDom`, and the user's Windows login name is `NTUser`.

Table 7) Home Directory shares.

Share Name	Complete Search Path	Command
~	/home/NTDom/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "~" **-path** "%d/%w" **-share-properties** homedirectory
~NTDom~NTUser	/home/NTDom/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "~%d~%w" **-path** "%d/%w" **-share-properties** homedirectory
~NTDom~NTUser	/home/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "~%d~%w" **-path** "%w" **-share-properties** homedirectory

~NTUser	/home/NTDom/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "~%w" **-path** "%d/%w" **-share-properties** *homedirectory*
NTUser	/home/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "%w" **-path** "%w" **-share-properties** homedirectory
CIFS.HOMEDIR	/home/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "CIFS.HOMEDIR" **-path** "%w" **-share-properties** homedirectory
~CIFS.HOMEDIR	/home/NTDom/NTUser	cluster:: **vserver cifs share create -vserver** vs0 **-share** "~CIFS.HOMEDIR" **-path** "%d/%w" **-share-properties** homedirectory

3. Create a user directory under the search path that corresponds to the share pattern specified when creating a share.

4.5 Common Failures, Diagnoses, and Solutions

This section describes common errors encountered during share access. It also provides the corresponding diagnostic procedures and appropriate solutions.

Share Path Cannot be Located

When you try to map a share, Windows presents the following error message:

```
System error 55 has occurred. The specified network resource or device is no longer available.
```

This error can occur for the following reasons:

- The share path associated with the share does not exist.

> **Best Practice**
>
> Always associate the physical directory before you create a CIFS share, and make sure that you have the correct directory path associated with the share.

- The share is created on a volume that does not exist.
- The maximum share name length is 255 bytes. If you define a share name that is longer than 255 bytes, the name is truncated.
- When accessing `homedir`, make sure that the search pattern and search path are specified correctly.

Access Denied

The following error message appears when you try to map a share:

```
System error 5 has occurred. Access is denied.
```

This error can occur for the follow reasons:

- The share-level ACL did not provide the user with adequate access permission.
- The user does not have enough file-level permission to access the share.
- The attached export policy rule does not include CIFS. To view the export policy, use this command:

```
cluster::> vserver export-policy show –vserver <vserver>
```

Authentication Failure

The following error message appears when you try to map a share:

```
Logon failure: unknown user name or bad password.
```

This means that you have entered the wrong username and/or password.

Internal Failure

When you encounter an internal failure, the following error message indicates that the storage server setup is wrong or is incomplete:

```
System error 59 has occurred. An unexpected network error occurred.
```

This error can be caused by:

- No name-mapping entry for the Windows user login.

  Check the event log for the `secd.cifsAuth.noNameMap` related message. Here is an example:

```
cluster::> event log show
5/2/2011 12:00:28   node2           DEBUG         secd.cifsAuth.noNameMap: vserver (vs1) CIFS
name to UNIX name mapping problem. User Authentication procedure failed!
 [    48] User 'CIFSDOM\Administrator' authenticated using NTLMv1 security
 [    48] Trying to map 'CIFSQA\Administrator' to UNIX user 'administrator' using implicit
mapping
 [    49] Connecting to NIS server 172.x.x.x
 [    59] Using a new connection to 172.x.x.x
 [    60] Name 'administrator' not found in UNIX authorization source NIS
 [    60] Name 'administrator' not found in UNIX authorization source LOCAL
 [    60] Could not get an ID for name 'administrator' using any NS-SWITCH authorization source
 [    60] Unable to map 'CIFSDOM\Administrator'. No default UNIX user defined.
 [    60] FAILURE: Unable to map Windows user 'CIFSDOM\Administrator' to appropriate UNIX user
```

- NBlade replay problem caused by inconsistency of share definition on the management interface and on NBlade.

  To check the share definition on NBlade, run the following command:

  `cluster::> diag NBlade cifs shares -vserver <vserver>`

  To check the share definition on the management interface, run the following command:

  `cluster::> vserver cifs shares –vserver <vserver> show`

5 Name Mapping

The Data ONTAP storage system is designed to support either CIFS credentials (NTFS), UNIX credentials (UNIX), or both (MIXED), depending on the volume security type. Both of the supported network file systems - CIFS and NFS - can be used to access the files on any of the volume types. In a multiprotocol environment, the name-mapping mechanism plays a key role in controlling the behavior that allows a user with credentials from one network file system type to be mapped to a user with credentials on another network file system type.

Mistakes in configuring name mapping can cause the following problems:

- If no name mapping entry was found, there is a high probability that the user will be denied access or will not even be able to log into the CIFS server.
- The user might be granted undesired access rights to the system.

Note: Group mapping is not supported in Data ONTAP. Even if you define a group mapping, the mapping entry is ignored under all conditions.

The name mapping procedure is accomplished in three sequential steps:

1. **Explicit name mapping.** Explicit name mapping can be defined either locally through the clusterwide name-mapping table or through LDAP. The name-mapping service switch can be defined by using the following command:

   ```
   cluster::> vserver modify -vserver <vserver> -fields nm-switch {file|ldap}
   ```

2. **Implicit name mapping.** When explicit name mapping does not find a matching entry, the system tries implicit name mapping.

   Suppose that the Windows domain for the CIFS server is called DOMAIN.COM. Then:

   – Windows user DOMAIN.COM\user1 is mapped to the UNIX user called user1.

   – UNIX user2 is mapped back to Windows user DOMAIN.COM\user2.

Best Practice

- Define an entry that maps Windows user *\root to a UNIX user nobody or pcuser to prevent the NT root user from being mapped to UNIX UID 0 (root user, superuser).
- Define an entry that maps the UNIX user administrator to a Windows Guest user to prevent spoofing the NT administrator account from UNIX.

3. **Default name mapping.** When both explicit and implicit name mapping fail, the default name mapping is used, if defined, as the last resort.

Best Practice

To prevent unexpected mapping, NetApp recommends setting up a default name mapping rule through `vserver cifs options` and only giving the Guest account to the default mapping account.

5.1 Explicit Local Name Mapping

Local name mapping is a rule-based clusterwide name mapping. The rules are defined by using regular expressions and are evaluated at the time of conversion, yielding different results based on the input name. (For more information about regular expressions, see Appendix A.)

Local name mapping entries are created in the name-mapping table. The entries are applied in the order in which they occur in the priority list. For example, a name mapping that occurs at position 2 in the priority list is applied before a name mapping that occurs at position 3. Each mapping direction (Kerberos-to-UNIX, Windows-to-UNIX, and UNIX-to-Windows) has its own priority list.

The following example creates a name mapping on a virtual server named vs0. This is a name mapping from UNIX to Windows at position 5 in the priority list; it maps the UNIX user user1 to the domain user Domain\User1 Users.

```
cluster::> vserver name-mapping create -vserver vs0 -direction unix-win -position 5
-pattern UnixUser1 -replacement "Domain\\NTUser1"
```

5.2 Explicit ldap-type Name Mapping

LDAP can be configured to provide a name-mapping service by modifying the LDAP client schema. Use the following command to modify the attribute name for the mapping Windows account:

```
cluster::> vserver services ldap client schema modify -windows-account-attribute <mapping
WindowsAccount attribute>
```

Finally, follow item 6 ("Name the service switch and setup") in section 3.1, Prerequisites, to configure the LDAP service.

5.3 Default Name Mapping

If both explicit and implicit name mapping fail, a default name mapping is used; a CIFS identity is mapped to an NFS identity with the same user name; and an NFS identity is mapped to a CIFS identity with the same user name of the home domain.

You can also set the default UNIX user for any unmapped CIFS users with the following option:

```
cluster::> vserver cifs options modify -vserver <vserver> -default-unix-user <default-unix-user>
```

5.4 Defensive Name Mapping

Best Practices
To prevent the possible unwanted behavior that can be caused by implicit name mapping and default name mapping, define the following two entries in the name-mapping table: • Define a `win-unix` mapping entry that maps Windows user "`*\*`" to an non-existing UNIX user, such as "`non-existing-unix-user` ". • Define a `unix-win` mapping entry that maps UNIX user "`*`" to an non-existing Windows user, such as "`non-existing-nt-user` ".

5.5 Common Mistakes

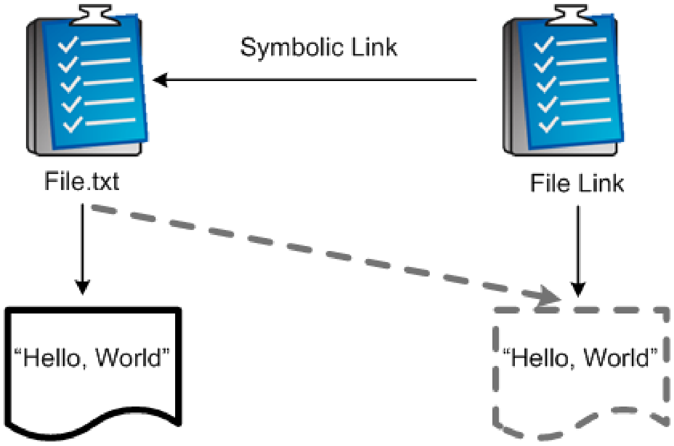
When you specify the replacement Windows user, be sure to enter a double back slash "\\" instead of a single one "\."

6 Symlink and Widelink

6.1 Symlink

A symlink (symbolic link or soft link) is a special type of file that contains a reference to another file or directory in the form of an absolute or relative path, which affects pathname resolution.

Figure 1) Symlink.



From a CIFS perspective, symbolic links are not allowed to leave a share for two reasons:

- Such a symlink would circumvent share-level security protection.

- The destination of a symlink depends on the way an NFS client has mounted its file systems. If the destination is not local to the system, the system would have to take the role of an NFS client itself to reach the destination.

There are two types of NFS symlink—relative and absolute.

Relative Symlinks

Relative symlinks always refer to an object within the Vserver namespace boundaries. For example:

```
foo_link->foo.txt
foo_link->../foo.txt
foo_link_dir->level2/foo_dir
```

Absolute Symlinks

Absolute symlinks refer to an object via a local mount point name that is resolved on the client side. Whether the symlink refers to an object within the Vserver namespace boundaries depends on the mounts on the clients. For example:

```
foo_link->/u/boyles/foo.txt
foo_link->/mnt/foo.txt
foo_link_dir->/mnt/level2/foo_dir
```

Where `/u` and `/mnt` might mount the same server where the link resides; or a they might mount a different server.

6.2 Widelink

Widelink is a feature that emulates Microsoft's DFS referral functionality. It allows DFS-enabled CIFS clients to resolve paths to locations that may not be contained inside a share. In fact, a widelink can allow path traversal to proceed from the original CIFS server also; hence the name widelink. This can work across Vservers.

To configure a widelink that travels outside the original CIFS server, follow these steps.

1. Mount an NFS share where you want to create a symlink.

   For example, on a UNIX box, the namespace is `/stuff` for the root of the NFS share. The directory is mounted on the UNIX box under `/mnt/stuff`:

```
[root@fedora mnt]# mount -t nfs 10.10.100.10:stuff /mnt/stuff
[root@fedora mnt]# cd stuff
```

2. Create a symlink under a mounted NFS share. The link must be in `/xxxx/` notation.

```
[root@fedora stuff]# ln -s /bing/ bingwide
[root@fedora stuff]# ls -la
total 28
drwxrwxrwx   3 root root   4096 Oct 30  2008 .
drwxr-xr-x  45 root root   4096 Oct 24 14:42 ..
drwxrwxrwx  11 root root   4096 Oct 30 15:05 .snapshot
lrwxrwxrwx   1 root root      8 Oct 30  2008 bingwide -> /bing/
drwxrwxrwx   2 bwc  users 16384 Oct 30 13:40 wallpapers
```

3. Create a symlink path mapping entry for the symlink created in step 2. The UNIX path must have both leading and trailing slashes.

   For example, a symlink path mapping entry created for a UNIX path `/bing` is translated to the path `\\CIFSsrv\user\bing` in the Data ONTAP system.

```
cluster::> vserver cifs symlink> create -vserver <vserver> -unix-path "/bing/"
```

```
-share-name "users" -cifs-path "bing" -cifs-server <CIFS server> -locality widelink
cluster::> vserver cifs symlink show
Virtual
Server    Unix Path   CIFS Server  CIFS Share   CIFS Path  Locality
--------  ----------  -----------  ----------   ---------  ---------
vserver   /bing/      CIFSsrv      user         bing       widelink
```

4. Enable the symlink property on the share you want to access from the Windows client. Suppose that symlink `bingwide` resides under the `stuff` share:

```
cluster::> vserver cifs shares modify -vserver <vserver> -share-name stuff -symlink-properties
enable
```

Note: NFS services should be running in conjunction with the CIFS services on the same Vserver. The symlinks that show up in CIFS must be created under NFS via a client computer.

5. Map the CIFS share as you normally would:

```
C:\WINDOWS>net use y: \\CIFSsrv\stuff /user:NTAP\bing
The command completed successfully.
```

The folder `bingwide` appears in the directory listing. Click that folder to be redirected to `\\CIFSsrv\user\bing`.

6.3 Enable Symlink Resolution

Symlink resolution in CIFS is enabled on a per-share basis. Enable it by using the following command:

```
cluster::> vserver cifs share modify -vserver <vserver name> -share-name <Share>
-symlink-properties enable
```

6.4 Common Failures

Due to the complexity of widelink setup, you may encounter the following failures while accessing the widelink object

1. Access denied.
   - Local symlink: Did you leave the source share scope to access the files and directories outside your source share?
   - Widelink:
     - Can the client find the target server?
       If not, check the DNS configuration.
     - Can the user authenticate into the target server?
     - Can the user access the redirected share on the target storage server?
2. Client cannot handle DFS referral response.

   Does the client have the capability to process DFS referral information?
3. Configuration-related issues.

   You can verify symlink and widelink configuration on NBlade by using the `path-mapping` command under diag privilege:

```
cluster::> set diag
Warning: These diagnostic commands are for use by NetApp personnel only.
Do you want to continue? {y|n}: y

cluster::*> diag nblade cifs path-mapping resolve –vserver <vserver> -unix-path <path>
```

7 Group Policy

Group Policy is a central component of Microsoft's change and configuration strategy starting in Windows 2000. With Group Policy, you can define users' environments and system configurations from one location. The settings you can control with Group Policy include environmental settings, user rights assignment, account policies, folder redirection, script assignment, security settings, and software distribution.

Group Policy is implemented by associating a Group Policy Object (GPO) with an Active Directory container, such as a site, domain, or organizational unit. A Group Policy Object is a collection of attribute settings. The settings associated with a GPO govern a user's or computer's configuration. Multiple containers can be linked to the same GPO, and a single container can have more than one linked GPO. If a GPO applies to one of these AD containers, then by default, all the settings in that GPO apply to every user and computer object in that container.

For more information on Group Policy, see http://technet.microsoft.com/en-us/library/cc736953(WS.10).aspx.

Clustered Data ONTAP 8.1 supports a limited set of GPOs:

- Refresh time interval
- Refresh time interval random offset
- Kerberos policies that are part of the security type of group policies

To enable Group Policy, use the following command:

```
cluster::> :vserver cifs group-policy modify –vserver <vserver> -status enabled
```

To force Group Policy refresh on a particular Vserver, use the following command:

```
cluster::> :vserver cifs group-policy update –vserver <vserver>
```

To show the Group Policies that have been applied to a particular Vserver, use the following command:

```
cluster::> :vserver cifs group-policy show-applied –vserver <vserver>
```

To show the Group Policies that were defined on the Active Directory for a particular CIFS server, use the following command:

```
cluster::> :vserver cifs group-policy show-defined –vserver <vserver>
```

8 Monitoring SMB Statistics

8.1 Statistics With Summary

Clusterwide Statistics:

```
cluster::> statistics show-periodic -node cluster:summary
cluster:summary: cluster: 8/4/2011 10:37:50
  cpu  cpu    total                     fcache             total    total data     data     data
cluster  cluster  cluster     disk     disk     pkts     pkts
  avg busy      ops  nfs-ops cifs-ops      ops spin-ops     recv     sent busy     recv     sent
busy     recv     sent     read    write     recv     sent
 ---- ---- -------- -------- -------- -------- -------- -------- -------- ---- -------- --------
------- -------- -------- -------- -------- -------- --------
  98%  98%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  90%  90%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  96%  96%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  98%  98%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  98%  98%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  98%  98%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  54%  54%        0        0        0        0        0   1.39KB     413B   0%       0B       0B
0%   1.22KB     230B   7.55KB   22.6KB       11        2
  97%  97%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  91%  91%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0
  98%  98%        0        0        0        0        0       0B       0B   0%       0B       0B
0%       0B       0B       0B       0B        0        0

...
```

Vserver-Wide Statistics:

```
cluster::> statistics show -node <node> -object cifs -instance vs0

Node: Node01
    Object.Instance.Counter                                     Value         Delta
    ----------------------------------------------- ------------- -------------
    cifs.vs0.instance_name                                        vs0
             -
    cifs.vs0.node_name
                                                               Node01
             -
    cifs.vs0.cifs_ops                                          25             -
    cifs.vs0.cifs_latency                                 10440us             -
    cifs.vs0.cifs_read_ops                                      0             -
    cifs.vs0.cifs_write_ops                                     0             -
    cifs.vs0.commands_outstanding                               0             -
    cifs.vs0.connections                                        1             -
    cifs.vs0.established_sessions                               1             -
    cifs.vs0.connected_shares                                   1             -
    cifs.vs0.open_files                                         0             -
    cifs.vs0.active_searches                                    0             -
    cifs.vs0.auth_reject_too_many                               0             -
13 entries were displayed.
```

8.2 Statistics at Intervals

```
cluster::> statistics show-periodic -node Node01 -object cifs -instance vs0 -interval 1
Node01: cifs.vs0: 8/4/2011 10:51:36
                                         cifs     cifs
auth_reject
 instance     node             cifs     read    write    commands             established
connected     open   active         too
     name     name cifs_ops  latency      ops      ops outstanding connections    sessions
shares    files searches        many
 -------- -------- -------- -------- -------- -------- ----------- ----------- ----------- ------
--- -------- -------- -----------
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
        0        0        0      0us        0        0           0           0           0
0        0        0           0
Node01: cifs.vs0: 8/4/2011 10:51:45
                                         cifs     cifs
auth_reject
 instance     node             cifs     read    write    commands             established
connected     open   active         too
     name     name cifs_ops  latency      ops      ops outstanding connections    sessions
shares    files searches        many
 -------- -------- -------- -------- -------- -------- ----------- ----------- ----------- ------
--- -------- -------- -----------
Minimums:
        0        0        0      0us        0        0           0           0           0
0        0        0           0
Averages for 9 samples:
        0        0        0      0us        0        0           0           0           0
0        0        0           0
Maximums:
        0        0        0      0us        0        0           0           0           0
0        0        0           0
```

8.3 CIFS Protocol Request Size

```
Node01::> statistics protocol-request-size show -node cpspectre8 -stat-type
cifs_read  cifs_write nfs2_read  nfs2_write nfs3_read  nfs3_write
Node01::> statistics protocol-request-size show -node cpspectre8 -stat-type cifs_read

Node:                    Node01
Stat Type:               cifs_read
                         Value       Delta
--------------           --------    ----------
Average Size:                   0             -
Total Request Count:            0             -
0-511:                          0             -
512-1023:                       0             -
1K-2047:                        0             -
2K-4095:                        0             -
4K-8191:                        0             -
8K-16383:                       0             -
16K-32767:                      0             -
32K-65535:                      0             -
64K-131071:                     0             -
```

```
128K - :                        0          -
```

8.4 CIFS Overall

```
cluster::> statistics show -node Node01 -object cifs

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    cifs.[proto_ctx:3].instance_name
                                                   [proto_ctx:3]
             -
    cifs.[proto_ctx:3].node_name                           Node01
             -
    cifs.[proto_ctx:3].cifs_ops                               474             -
    cifs.[proto_ctx:3].cifs_latency                       52728us             -
    cifs.[proto_ctx:3].cifs_read_ops                            0             -
    cifs.[proto_ctx:3].cifs_write_ops                           0             -
    cifs.[proto_ctx:3].commands_outstanding                     0             -
    cifs.[proto_ctx:3].connections                              0             -
    cifs.[proto_ctx:3].established_sessions                     0             -
    cifs.[proto_ctx:3].connected_shares                         0             -
    cifs.[proto_ctx:3].open_files                               0             -
    cifs.[proto_ctx:3].active_searches                          0             -
    cifs.[proto_ctx:3].auth_reject_too_many                     0             -
    cifs.[proto_ctx:4].instance_name
                                                   [proto_ctx:4]
             -
    cifs.[proto_ctx:4].node_name                           Node01
             -

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    cifs.[proto_ctx:4].cifs_ops                                 0             -
    cifs.[proto_ctx:4].cifs_latency                           0us             -
    cifs.[proto_ctx:4].cifs_read_ops                            0             -
    cifs.[proto_ctx:4].cifs_write_ops                           0             -
    cifs.[proto_ctx:4].commands_outstanding                     0             -
    cifs.[proto_ctx:4].connections                              0             -
    cifs.[proto_ctx:4].established_sessions                     0             -
    cifs.[proto_ctx:4].connected_shares                         0             -
    cifs.[proto_ctx:4].open_files                               0             -
    cifs.[proto_ctx:4].active_searches                          0             -
    cifs.[proto_ctx:4].auth_reject_too_many                     0             -
    cifs.vs1.instance_name                                   vs1
             -
    cifs.vs1.node_name                                     Node01
             -
    cifs.vs1.cifs_ops                                         474             -
    cifs.vs1.cifs_latency                                 52728us             -
    cifs.vs1.cifs_read_ops                                      0             -

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    cifs.vs1.cifs_write_ops                                     0             -
    cifs.vs1.commands_outstanding                               0             -
    cifs.vs1.connections                                        0             -
    cifs.vs1.established_sessions                               0             -
    cifs.vs1.connected_shares                                   0             -
    cifs.vs1.open_files                                         0             -
    cifs.vs1.active_searches                                    0             -
    cifs.vs1.auth_reject_too_many                               0             -
39 entries were displayed.
```

8.5 SMB1 and SMB2 Specific Statistics

```
cluster::> statistics show -node Node01 -object smb1

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    smb1.[proto_ctx:003].instance_name
                                                 [proto_ctx:003]
             -
    smb1.[proto_ctx:003].node_name
                                                           Node01
             -
    smb1.[proto_ctx:003].ops                                    0             -
    smb1.[proto_ctx:003].latency                              0us             -
    smb1.[proto_ctx:003].read_class_ops                         0             -
    smb1.[proto_ctx:003].write_class_ops                        0             -
    smb1.[proto_ctx:003].commands_outstanding                   0             -
    smb1.[proto_ctx:003].established_sessions                   0             -
    smb1.[proto_ctx:003].connected_shares                       0             -
    smb1.[proto_ctx:003].open_files                             0             -
    smb1.[proto_ctx:003].active_searches                        0             -
    smb1.[proto_ctx:004].instance_name
                                                 [proto_ctx:004]
             -
    smb1.[proto_ctx:004].node_name
                                                           Node01
             -

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    smb1.[proto_ctx:004].ops                                    8       0/s:20s
    smb1.[proto_ctx:004].latency                              0us      -1805026
    smb1.[proto_ctx:004].read_class_ops                         0             -
    smb1.[proto_ctx:004].write_class_ops                        0             -
    smb1.[proto_ctx:004].commands_outstanding                   0             -
    smb1.[proto_ctx:004].established_sessions                   0             -
    smb1.[proto_ctx:004].connected_shares                       0             -
    smb1.[proto_ctx:004].open_files                             0             -
    smb1.[proto_ctx:004].active_searches                        0             -
    smb1.vs0.instance_name                                   vs0
             -
    smb1.vs0.node_name
                                                           Node01
             -
    smb1.vs0.ops                                                8       0/s:17s
    smb1.vs0.latency                                          0us      -1805026
    smb1.vs0.read_class_ops                                     0             -

Node: cpspectre8-01-02
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    smb1.vs0.write_class_ops                                    0             -
    smb1.vs0.commands_outstanding                               0             -
    smb1.vs0.established_sessions                               0             -
    smb1.vs0.connected_shares                                   0             -
    smb1.vs0.open_files                                         0             -
    smb1.vs0.active_searches                                    0             -
33 entries were displayed.


cluster::> statistics show -node Node01 -object smb2

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    smb2.[proto_ctx:003].instance_name
                                                 [proto_ctx:003]
             -
    smb2.[proto_ctx:003].node_name
```

```
                                                           Node01
             -
    smb2.[proto_ctx:003].ops                                    0             -
    smb2.[proto_ctx:003].latency                              0us             -
    smb2.[proto_ctx:003].read_class_ops                         0             -
    smb2.[proto_ctx:003].write_class_ops                        0             -
    smb2.[proto_ctx:003].read_class_latency                   0us             -
    smb2.[proto_ctx:003].write_class_latency                  0us             -
    smb2.[proto_ctx:003].commands_outstanding                   0             -
    smb2.[proto_ctx:003].established_sessions                   0             -
    smb2.[proto_ctx:003].connected_shares                       0             -
    smb2.[proto_ctx:003].open_files                             0             -
    smb2.[proto_ctx:003].active_searches                        0             -
    smb2.[proto_ctx:004].instance_name
                                                 [proto_ctx:004]
             -

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    smb2.[proto_ctx:004].node_name
                                                           Node01
             -
    smb2.[proto_ctx:004].ops                                   25             -
    smb2.[proto_ctx:004].latency                          10440us             -
    smb2.[proto_ctx:004].read_class_ops                         0             -
    smb2.[proto_ctx:004].write_class_ops                        0             -
    smb2.[proto_ctx:004].read_class_latency                   0us             -
    smb2.[proto_ctx:004].write_class_latency                  0us             -
    smb2.[proto_ctx:004].commands_outstanding                   0             -
    smb2.[proto_ctx:004].established_sessions                   1             -
    smb2.[proto_ctx:004].connected_shares                       1             -
    smb2.[proto_ctx:004].open_files                             0             -
    smb2.[proto_ctx:004].active_searches                        0             -
    smb2.vs0.instance_name                                   vs0
             -
    smb2.vs0.node_name
                                                           Node01
             -

Node: Node01
    Object.Instance.Counter                                 Value         Delta
    ----------------------------------------------- ------------- -------------
    smb2.vs0.ops                                               25             -
    smb2.vs0.latency                                      10440us             -
    smb2.vs0.read_class_ops                                     0             -
    smb2.vs0.write_class_ops                                    0             -
    smb2.vs0.read_class_latency                               0us             -
    smb2.vs0.write_class_latency                              0us             -
    smb2.vs0.commands_outstanding                               0             -
    smb2.vs0.established_sessions                               1             -
    smb2.vs0.connected_shares                                   1             -
    smb2.vs0.open_files                                         0             -
    smb2.vs0.active_searches                                    0             -
39 entries were displayed.
```

9 Appendix A: Regular Expressions for Name Mapping

Rules act as templates through which users are translated. Rules can be used in pattern and/or replacement fields when a local name-mapping entry is created. For example:

```
NETAPP\\(*) -> \1
```

Through this rule, the Windows user named "NETAPP\user01" maps to the UNIX user "user01". This is explained in detail later in this section.

A clustered Data ONTAP system keeps a domain configuration and a set of conversion rules for each virtual server. Each rule consists of two pieces, a pattern and a replacement. In the example above, the left side of the mapping is the pattern, and the right side is the replacement.

Conversions start at the beginning of the virtual server's map list and perform a substitution based on the first matching rule. The pattern is a UNIX-style regular expression. The replacement is a string containing escape sequences representing subexpressions from the pattern, as in the UNIX `sed` program.

Regular expressions are not case sensitive when mapping from Windows to UNIX. However, they are case sensitive for Kerberos-to-UNIX and UNIX-to-Windows mappings.

The backslash (\) character is used as an escape sequence in regular expression patterns and replacements. If a backslash is desired in the input or output, it must be escaped with another backslash, as in the example above.

The caret (^), underscore (_), and ampersand (&) characters can be used as prefixes for characters in replacement patterns. These characters specify translations of the output name into uppercase, lowercase, and initial-case transformations, respectively. For example:

- If the initial pattern is (.+) and the replacement pattern is \1, then the string jOe is mapped to jOe (no change).
- If the initial pattern is (.+) and the replacement pattern is _1, then the string jOe is mapped to joe.
- If the initial pattern is (.+) and the replacement pattern is \^1, then the string jOe is mapped to JOE.
- If the initial pattern is (.+) and the replacement pattern is \&1, then the string jOe is mapped to Joe.
- If the character following a backslash-underscore (_), backslash-caret (\^), or backslash-ampersand (\&) sequence is not a digit, then the character following the backslash is used verbatim.

The following example converts any Windows user in the Windows domain named ENG into a UNIX user with the same name in NIS:

```
Vserver       Direction     Pattern            Replacement
Vs0           win-unix      ENG\\(.+)        \1
```

The double backslash (\\) in the pattern matches a single backslash in the source name. The parentheses denote a subexpression but do not match any characters themselves. The period matches any single character. The plus sign matches one or more of the previous expressions. In this example, you are matching ENG\ followed by one or more of any character. In the replacement, \1 refers to the first subexpression matched. Assuming the Windows user ENG\jones, the replacement evaluates to jones; that is, the portion of the name following ENG\.

10 Appendix B: Diagnosis and Tracing

Clustered Data ONTAP implements CIFS features in various modules. Each module offers different levels of diagnostic and tracing functionalities. This section introduces commonly used commands for diagnostic purpose during a failure event or maintenance.

Low-level or detailed diagnostic utilities are mostly available for system administrators with diag privilege. Use them with caution. NetApp recommends contacting its support team for advice. To access those commands, you must be in diag mode:

```
cluster::> set diag
Warning: These diagnostic commands are for use by NetApp personnel only.
Do you want to continue? {y|n}: y
```

10.1 Event Logging

Events provide information about the operations that occur on the system. An event consists of the following elements:

- Message name, sequence number, and time stamp
- Severity level
- Description
- Corrective action, if applicable

The event log is a mostly used to check a system for points of failures. It is easy to search message logs for instances of particular events based on the display criteria.

To display the event log, use `event log show` with optional parameters:

```
cluster::> event log show ?
  [ -detail | -detailtime | -instance | -fields <fieldname>, ... ]
  [[-node] <nodename>]                 Node
  [[-seqnum] <Sequence Number>]        Sequence#
  [ -time <"MM/DD/YYYY HH:MM:SS"> ]    Time
  [ -severity {EMERGENCY|ALERT|CRITICAL|ERROR|WARNING|NOTICE|INFORMATIONAL|DEBUG} ]
                                       Severity
  [ -source <text> ]                   Source
  [ -messagename <Message Name> ]      Message Name
  [ -event <text> ]                    Event
```

Examples:

- To show all the events with selected fields:

```
cluster::event log show -node n1 -fields seqnum,time,severity,source,messagename
node  seqnum time                            severity                  source  messagename
------- ----------- --------------------------- ---------------------    ---------  ----------
------------------------------------------
n1      4739      "2/14/2011 15:44:09" INFORMATIONAL mgwd   mgmtgwd.jobmgr.jobcomplete.success
n1      4738      "2/14/2011 15:44:09" INFORMATIONAL mgwd   mgmtgwd.jobmgr.jobstart
n1      4690      "2/14/2011 15:38:57" ERROR                  mgwd   mgmt.snapmir.schd.trans.fail
n1      4683      "2/14/2011 15:38:10" WARNING             mgwd   mgmt.snapmir.schd.trans.overrun
n1      4668      "2/14/2011 15:35:57" DEBUG                  mgwd   mgmt.snapmir.abnormal.abort
n1      4616      "2/14/2011 15:30:02" DEBUG                  secd     secd.nfsAuth.noUnixCreds
```

- To show detailed information about an individual event with a specific sequence number:

```
cluster::event log show -node n1 –seqnum 4616
                Node: n1
           Sequence#: 4616
                Time: 2/14/2011 15:30:02
            Severity: DEBUG
              Source: secd
        Message Name: secd.nfsAuth.noUnixCreds
               Event: secd.nfsAuth.noUnixCreds: vserver (vs1) Cannot determine UNIX identity.
Acquire UNIX Credentials procedure failed!
  [  0 ms]  ID 65534 not found in UNIX authorization source LOCAL
  [     0]  Could not get credentials for ID 65534 using any NS-SWITCH authorization source
**[     0]  FAILURE: Unable to retrieve credentials for UNIX user with UID 65534
```

10.2 NBlade CIFS

The CIFS protocol runs on the NBlade networking blade. System administrators set up CIFS servers and related features through the management console interface. That setup information is populated into NBlade during:

- System startup
- Feature modification
- Logical interface migration

CIFS requests and responses are processed on the NBlade workspace. Depending on the request type, the CIFS module contacts SecD for security-related requests and DBlade for data-access-related requests.

A series of diagnostic routines for NBlade CIFS protocol implementation is embedded under the diag `NBlade cifs` command directory. These commands show an overall picture of the NBlade view of the CIFS configuration, cached credentials, and current working state.

1. Server command. This command returns information about a specific CIFS virtual server or, if no Vserver is specified, about all configured CIFS servers:

```
cluster::*> diag NBlade cifs server show
 There are 1 Virtual Servers:
       Virtual Server ID:     1
       Server Type:           Kerberos
       Name:                  VS0
       Domain Name:           CIFSDOM
       MTAP:                  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
       Node Type:             B
       WINS Server IP:        0.0.0.0
       Machine SID:           S-1-000000000005-21-1407423728-2963865486-1834115207
       Local SID:             S-1-000000000005-21-1407423728-2963865486-1834115207-33169
       Domain Controllers:    0
       IP Addresses:          Count = 1
       IP (0):                100.10.100.10
```

2. Share command. This command lists share information for all CIFS shares on a given Vserver. It is used to verify the consistency of share definitions on the management interface and on NBlade. Inconsistencies usually indicate a replay problem on NBlade.

```
cluster::*> diag nblade cifs shares -vserver vs0
 Virtual server vs0 has 3 shares:
  Share:  test
      Path:               /test
      Type:               0
      Oplocks:            on
      ChangeNotify:       on
      Dynamic Share:      off
      Dynamic Path:       off
      Browsable:          on
      Show Snapshot:      off
      File Umask(octal):  0
      Dir Umask(octal):   0
      Symlinks:           off
  Share:  ipc$
      Path:
      Type:               3
      Oplocks:            off
      ChangeNotify:       off
      Dynamic Share:      off
      Dynamic Path:       off
      Browsable:          on
      Show Snapshot:      off
      File Umask(octal):  0
      Dir Umask(octal):   0
      Symlinks:           off
  Share:  admin$
      Path:
      Type:               3
      Oplocks:            off
      ChangeNotify:       off
      Dynamic Share:      off
      Dynamic Path:       off
      Browsable:          on
      Show Snapshot:      off
      File Umask(octal):  0
      Dir Umask(octal):   0
      Symlinks:           off
```

3. Interface command. This command lists all configured data interfaces for a given Vserver. It lists only the data interfaces, not the cluster interfaces.

```
cluster::*> diag nblade cifs interfaces -vserver vs0
Interfaces for Virtual Server vs0
   Interface ID: 1026
     IP Address: 172.1.0.11

   Interface ID: 1025
     IP Address: 172.1.0.10
```

4. Path mapping commands. These commands are mostly used to verify the widelink configuration and to check the correctness of a widelink setup. The set consists of two subcommands, `show` and `resolve`.

   a) The `show` command lists all path mappings for a given Vserver.

      Here is a wildlink setup through the management interface:

```
cluster::> vserver cifs symlink show
Virtual
Server    Unix Path   CIFS Server  CIFS Share   CIFS Path  Locality
------    ---------   -----------  -----------  ---------- ------------
vs0       /bing/      CIFSsrv      user         bing       widelink
```

The `diag` command shows the corresponding CIFS path, `\\CIFSsrv\user\bing\,` for a UNIX path called `bing`.

```
cluster::*> diag nblade cifs path-mapping show –vserver vs0
VserverId    Type    Unix Path                        Cifs Path
----------   ------  -------------------------------  ------------------------------
1            1       /bing/                           \CIFSsrv\users\bing\
```

b) The `resolve` command helps to resolve a path mapping from a UNIX path to a CIFS path.

```
cluster::*> diag secd path-mapping resolve -vserver vs0 -unix-path /bing/Documents
VserverId     Unix Path                          Cifs Path
------------ --------------------------------- ---------------------------------------
1            /bing/                              \CIFSsrv\user\bing\
```

10.3 Authentication

Authentication can be a lengthy process, usually involving interactions between multiple parties, such as KDC, NIS, LDAP, and DC. This process can be hard to diagnose from the end-user's side. A clustered Data ONTAP system provides a set of authentication commands that allow the system administrator to isolate the authentication problem from a storage server's perspective.

Authentication-related commands are under the `diag secd authentication` command directory.

1. Authenticate a CIFS user without client involvement. This command can be used to test the authentication process between the system and the domain controller. Upon successful authentication, a user credential is displayed.

```
cluster::*> diag secd authentication login-cifs -node n1 -vserver vs0 -user cifsdom\administrator
Enter the password:
Windows User: administrator Domain: cifsdom Privs: a7
    Primary Grp: S-1-5-21-1407423728-2963865486-1834115207-513
          Domain: S-1-5-21-1407423728-2963865486-1834115207 Rids: 500, 520, 513, 22226, 26625,
1842, 512, 519, 518, 8323, 1645, 1648, 1644, 1647, 1003
          Domain: S-1-1 Rids: 0
          Domain: S-1-5 Rids: 11, 2
           Unix ID: 0, GID: 1
              Flags: 0
      Domain ID: 0
     Other GIDs: 12, 9, 5, 0, 6, 7, 3, 2, 4, 8
Authentication Succeeded.
```

2. Show user's credential based on a CIFS SID or a UNIX UID. A user's credential determines that user's privileges on the storage system, as well as access rights to files and directories in the system. This command can be very useful when diagnosing problems in the authorization process:

```
cluster::*> diag secd authentication show-creds
```

3. Translate a Windows SID to a UNIX ID or a UNIX ID to a Windows SID. This is useful for diagnosing name-mapping and authorization-related problems:

```
cluster::*> diag secd authentication sid-to-uid
cluster::*> diag secd authentication uid-to-sid
```

4. Translate from various types of names to its identifier. This is useful for finding naming-service-related issues:

```
cluster::*> diag secd authentication translate
```

10.4 Connection Cache

SecD extensively caches connections to external servers, which is a key to improving the performance and robustness of the Data ONTAP system.

This mechanism manages the cache centrally and greatly reduces number of calls to external servers. The ability to display and/or clear the connection cache helps you to diagnose network connectivity-related problems.

1. Display active connections for a specific Vserver:

```
cluster::*> diag secd connections show -node <node> -vserver <vserver>
```

2. Display connections of specific types—LSA, Netlogon, LDAP, NIS:

```
cluster::*> diag secd connections show -node <node> -vserver <vserver> -type <Cache type>
```

Example:

```
cluster::*> diag secd connections show -node n1 -vserver vs0 -type netlogon
[Cache: NetLogon/cifsqa.lab.netapp.com - Hits: 2, misses: 4, average retrieval: 64.00ms]
 + Rank: 01 - Server: 100.10.100.10 (cifsdc-1.cifsdom.corp.com)
              Connected through the 172.17.208.200 interface, 38.5 mins ago
              Used 2 time(s), and has been available for 2293 secs
              RTT in ms: mean=0.00, min=0, max=0, med=0, dev=0.00 (0.0 mins of data)
```

10.5 Security Cache

SecD extensively caches data from as well as to external servers. This increases performance and serves higher data availability to the clustered storage system.

The `diag secd cache` command set allows system administrators to view, clear, and exercise limited control over various types of security caches. For example, you can manage the number of maximum entries and the lifetime of an entry for different types of caches. Note that a smaller value for the lifetime of a cache affects the performance of related operations, because it takes SecD time to gather the information over the network resources. On the other hand, a larger value for the lifetime of a cache could affect the security accuracy, because SecD might still access the old or cached security data instead of updated information over the network resources.

- To show a cache configuration:

```
cluster::*> diag secd cache show-config -node <node> -cache-name <cache_name>
```

- To set a cache configuration:

```
cluster::*> diag secd cache set-config –node <node> -cache-name <cache_name> -max-entries <max-
entries> -life-time <life-time>
```

- To view or dump a cache configuration, use the following command. After the `dump` command is issued, the related cache content is dumped to a log file called `secd.log`, located at `/mroot/etc/log/mlog/` in FreeBSD workspace.

```
cluster::*> diag secd cache dump –node <node> -cache-name <cache_name>
```

Example: To dump an SID to a name cache:

```
cluster::*> diag secd cache dump –node n1 -cache-name sid-to-name
```

- To clear all the entries of a cache:

```
cluster::*> diag secd cache clear –node <node> -cache-name <cache_name>
```

Here is a list of commonly accessed caches for CIFS use:

- **ad-to-netbios-domain.** A mapping from a fully qualified Active Directory name to the corresponding NetBIOS name. This information is retrieved as needed from Active Directory.
- **netbios-to-ad-domain.** The reverse of the AD-to-NetBIOS-domain cache.
- **name-to-sid**: A mapping from a Windows user, group, or domain name to the matching Windows SID. This information is retrieved from the domain controllers.
- **sid-to-name.** The reverse of the name-to-sid cache.

10.6 Security Log

Security-log-related commands are used to manage the levels of messages logged in the `secd.log` file.

1. To show the current level of messages logged in `secd.log`, use the following command. To log each function name that the code traverses, set the `enter/exit` field to `On`.

```
cluster:: *> diag secd log show
Log Options
------------------------------------------------
Log level:                              DEBUG
Function enter/exit logging:  OFF
```

2. Change the log level and turn the `enter/exit` logging function on or off:

```
cluster::*> diag secd log set ?
   [-node] <nodename>         *Node
  [[-level] <log_level>]      *Log Level
  [ -enter-exit {on|off} ]    *on/off
```

Example:

```
cluster::*> diag secd log set -level debug -enter-exit on
Setting log level to DEBUG
Setting enter/exit to ON
```

10.7 Security Tracing

Security tracing is a mechanism to log RPCs when you need to see what SecD is doing even when the RPCs are succeeding. By default, SecD automatically logs failed RPCs. Security tracing also allows RPCs to be logged based on criteria such as severity level, security modules, and so on.

To access the security-related log file, exit to the BSD prompt. The log file is located at `/mroot/etc/log/mlog/secd.log`.

All security-trace-related commands are under `the diag secd trace` command directory.

```
cluster::*> diag secd trace ?
  clear                    *Clear Trace Options
  set                      *Set Trace Options
  show                     *Show Trace

Here is an example of steps to set and show a trace that traces everything:
```

1. Set the security trace by using the `-trace-all` option, which traces all RPCs, regardless of success or failure:

```
cluster::*> diag secd trace set -node n1 -trace-all yes
Trace spec set successfully.
```

2. View the security trace setup:

```
cluster::*> diag secd trace show –node n1
Trace Spec
-----------------------------------------------------
TraceAll:                     Tracing all RPCs
```

3. Now any RPCs called into the SecD get logged. Try to show this with a previous configuration RPC:

```
cluster::*> diag secd configuration query -node n1 -source-name name-mapping
         vserver: 2
            type: user
       direction: win-unix
        position: 1
          pattern: cifsqa\\administrator
      replacement: root

          vserver: 2
```

```
             type: user
        direction: unix-win
         position: 1
          pattern: root
      replacement: cifsqa\\Administrator
```

4. This produces the following log output. Notice that the tracing match is on `All`.

```
.-----------------------------------------------------------------------------------------------
.
|                                  TRACE MATCH
               |
| RPC secd_rpc_config_source_admin_op succeeded and is being dumped because of     |
|                              a tracing match on:
       |
|                                      All
               |
|                    RPC received at Mon May  9 15:25:26 2011
       |
' -----------------------------------------------------------------------------------------------
`
| [000.000.036]  debug:  Worker Thread 34369186320 processing RPC
703:secd_rpc_config_source_admin_op with request ID:21 which sat in the queue for 0 seconds.  {
in run() at server/secd_rpc_server.cpp:1459 }
| [000.000.087]  debug:  Got rows: '0,,0,,0,,0,,0,,0,,'  { in
secd_rpc_config_source_admin_op_1_svc() at configuration_manager/secd_rpc_config.cpp:535 }
| [000.000.112]  debug:  Received config source admin op 'query' for source name 'NameMapping'  {
in secd_rpc_config_source_admin_op_1_svc() at configuration_manager/secd_rpc_config.cpp:542 }
| [000.000.313]  debug:  Looking for source with name 'NameMapping'  { in
getConfigSourceFromName() at configuration_manager/secd_configuration_manager.cpp:1646 }
| [000.000.338]  debug:  Admin GENERAL for source 'secd_namemappings_db_view'  { in admin() at
../secd/include/secd_configuration_sources.h:414 }
| [000.000.376]  ERR  :  7009 in getElementValueAsStr() at ../secd/include/secd_common_types.h:86
| [000.000.411]  ERR  :  7009 in getElementValueAsStr() at ../secd/include/secd_common_types.h:86
| [000.000.432]  debug:  Querying config source 'NameMapping' (with 4 rows of data) by keys
vserver id: '0', direction: '', position: '0', type: ''  { in query() at
configuration_manager/secd_configuration_sources.cpp:3829 }
| [000.000.466]  debug:  Admin op query got 4 records  { in adminOpQuery() at
../secd/include/secd_configuration_sources.h:376 }
| [000.000.584]  debug:  Adding row with 6 fields to query results  { in adminOpQuery() at
../secd/include/secd_configuration_sources.h:379 }
| [000.000.686]  debug:  Adding row with 6 fields to query results  { in adminOpQuery() at
../secd/include/secd_configuration_sources.h:379 }
| [000.000.791]  debug:  Adding row with 6 fields to query results  { in adminOpQuery() at
../secd/include/secd_configuration_sources.h:379 }
| [000.000.898]  debug:  Adding row with 6 fields to query results  { in adminOpQuery() at
../secd/include/secd_configuration_sources.h:379 }
| [000.001.014]  debug:  SecD RPC Server sending reply to RPC 703:
secd_rpc_config_source_admin_op  { in secdSendRpcResponse() at server/secd_rpc_server.cpp:1355 }
.-----------------------------------------------------------------------------------------------
-----------.
|                   RPC completed at Mon May  9 15:25:26 2011                              |
|        End of log for successful RPC secd_rpc_config_source_admin_op.           |
'-----------------------------------------------------------------------------------------------
```

10.8 Network Packet Tracing

Taking network traces on the Data ONTAP storage system can be very helpful in uncovering problems with communication to external servers. This section describes how to take a network trace. To capture a packet trace on the system:

1. Start the trace:

```
cluster::> node run -node <node name> -command pktt start all
```

   This captures all traffic on all interfaces into a 128K circular buffer.

2. Cause the failure:

   Perform the action that attempts the external network access.

3. Write the trace:

```
cluster::> node run -node <node name> -command pktt dump all
```

   This writes all captured data to files named `/mroot/<interface>_<yyyymmdd>_<hhmmss>.trc`. If files with the same name already exist, they are overwritten. The data is in the tcpdump format.

4. Stop the trace:

```
cluster::> node run -node <node name> -command pktt stop all
```

   This stops tracing and flushes all data from the circular buffer.

5. Copy the trace to the working directory through the BSD shell:

```
cluster::> exit
node1% mount
node1% chmod 777 /mroot/*.trc
node1% cp /mroot/*.trc <your trace directory>
```

11 Conclusion

Although inspired by NetApp’s experience with Data ONTAP 7G, the all-new CIFS implementation in clustered Data ONTAP 8.1 been completely rearchitected. Redundancies caused by legacy patchwork and updates have been removed, and the new structure is designed to provide high scalability in a cluster system. The new CIFS architecture offers reliability through multiple check points and a built-in retry mechanism for robustness upon unexpected failures. CIFS servers are no longer bound to a single network interface; they can now be associated with multiple virtual interfaces that are located on different cluster controllers. This design also means that the cluster now has the ability to provide continuous CIFS server operation and related services.

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

NetApp®

www.netapp.com